**REMARKS**

The claims are claims 1 and 7.

Claim 1 has been amended to incorporate the limitations of canceled claim 4. Claim 7 has been amended to incorporate the limitations of canceled claim 9. Claims 2 to 6 and 8 to 11 are cancelled.

Claims 4 and 7 (reciting subject matter now recited in respective claims 1 and 7) were rejected under 35 U.S.C. 103(a) as made obvious by the combination of Simar et al EPO Published Patent Application No. 0 855 648 and Heishi et al U.S. Patent No. 6,324,639.

Claims 1 and 7 recite subject matter not made obvious by the combination of Simar et al and Heishi et al. Claim 1 recites each multiplexer has "exactly two data inputs, a first data input receiving an entire instruction from a predetermined section of said first latch and a second data input receiving an entire instruction from a corresponding section of said second latch" and selects at its output "either said entire instruction from said section of said first latch, said entire instruction from said section of said second latch, or no instruction." Claim 1 further states that the dispatch control circuit controls the multiplexers "via said control inputs according to the execute packets determined by only said p-bits." Claim 7 recites "selecting only either an entire instruction from a predetermined section of said first latch, an entire instruction from a corresponding section of said second latch, or no instruction, dependent upon only said p-bit from each instruction stored in said first latch and each instruction stored in said second latch."

These recitations differ from the teaching in Heishi et al in several ways. The "exactly two data inputs" language in claim 1 and the "selecting only" language of claim 7 differs from the

teachings of Heishi et al. Heishi et al fails to teach that his selectors 224a, 224b, 224c and 224d select instructions only from a predetermined section of the first latch and a corresponding section of the second latch. Figure 8 of Heishi et al clearly shows that each selector 224a, 224b, 224c and 224d is connected at all sections A0, A1, A2 and A3 of instruction buffer A 221 and all sections B0, B1, B2 and B3 of instruction buffer B 222. As a consequence each selector 224a, 224b, 224c and 224d of Figure 8 of Heishi et al requires six inputs, one input from each section A0, A1, A2, A3 of instruction buffer 221 and one input from each section B0, B1, B2 and B3 of instruction buffer 222. In contrast, the multiplexors recited in claims 1 and 7 require only two inputs, one from a single section of latch stage 910 and one from a corresponding single section of latch stage 911. Thus this invention accomplishes a similar purpose with much simpler hardware. Accordingly, claims 1 and 7 are not made obvious by the combination of Simar et al and Heishi et al.

The combination of Simar et al and Heishi et al fails to make obvious the multiplexer selection of no instruction. The examples of Heishi et al at column 14, lines 1 to 62 and illustrated in Figures 9A, 9B, 9C, 9D, 9E, 9F, 10A, 10B, 10C, 10D and 10E show that all selectors 224a, 224b, 224c and 224d select an instruction from either instruction buffer A 221 or instruction buffer B 222 each time a selection is made by any of the multiplexers. These selections are illustrated in Figure 9D (column 14, lines 27 to 31), Figure 10A (column 14, lines 44 to 48) and Figure 10D (column 14, lines page 53 to 58). Selectors 224a, 224b, 224c and 224d each select an instruction to be transferred to instruction register 23 even for instances when all such instructions cannot be dispatched. In this application selection of no instruction is the same as a coding of a no operation instruction according to Simar et al and

results in no instruction dispatch. Heishi et al states at column 15, lines 27 to 36:

> "The instruction issuing control unit 31 refers to the parallel execution boundary information f10 and the format information f11 of the units stored in the instruction register A231 and the instruction register B232, and judges which is the final unit that should be outputted from the instruction register 23 in this cycle. Based on this information, the instruction issuing control unit 31 outputs control signals (no-operation instruction flags) that show whether the decoding by the second instruction decoder 34 and third instruction decoder 35 should be invalidated."

Thus Heishi et al teaches another structure instruction issuing control unit 31 controlling the issue of no-operation control signals to first instruction decoder 33, second instruction decoder 34 and third instruction decoder 35 than recited in claims 1 and 7. Accordingly, claims 1 and 7 are not made obvious by the combination of Simar et al and Heishi et al.

Claim 1 recites the dispatch circuitry operates to "dispatch each instruction of said selected execute packet to a functional unit corresponding to said instruction type of said instruction." Claim 7 similarly recites "dispatching each instruction within the determined execute packet to one of a second plurality of execution units dependent upon an instruction type of the instruction." This subject matter corresponds to the cross point circuitry 930 illustrated in Figure 9. The Program Decode entry of Table 8 on page 21 of the original application states "Next execute packet in fetch packet determined and sent to the appropriate functional units to be decoded." Thus the dispatch circuitry of claim 1 and the dispatching step of claim 7 provide routing of instructions selected by multiplexors 920(0) to 920(7) to "appropriate functional units" "dependent upon an instruction type of the instruction." This function of the dispatch circuitry and the

dispatching step is illustrated in Figure 8, where: multiply instructions 820(0) and 820(1) are dispatched to respective functional units M1 and M2; add instructions 820(2) and 820(3) are dispatched to respective functional units L1 and L2; and store instructions 820(4) and 820(5) are dispatched to respective functional units D1 and D2. First instruction decoder 33, second instruction decoder 34 and third instruction decoder 35 of Heishi et al fail to perform this action. Heishi et al states at column 15, lines 9 to 14:

> "As shown in FIG. 11, the position in the instruction register 23 to which a unit is transferred is unequivocally determined by its position in the unit queue. This means, for example, that the first unit in the queue will be transferred to the instruction register A231 and the second unit will be transferred to the instruction register B232."

Figure 4 of Heishi et al illustrates three lines coming from instruction decoder 32, presumably from first instruction decoder 33, second instruction decoder 34 and third instruction decoder 34, respectively. These three outputs couple to respective R1 Bus and first calculating unit 44, R2 Bus and second calculating unit 45 and R3 bus and third calculating unit 46. Heishi et al states at column 12, lines 21 to 25 state:

> "The first calculating unit 44, the second calculating unit 45, and the third calculating unit 46 each include an ALU (arithmetic logic unit) and multiplier that perform calculations on two pieces of 32-bit data, as well as a barrel shifter that performs shift operations."

Thus each of these calculating units has the same hardware and can execute any instruction. Accordingly, the functional unit to which an instruction is dispatched in Heishi et al depends entirely on its position in the original instruction stream. This teaching of Heishi et al is contrary to the above quoted recitations of claims

1 and 7 which dispatch instructions to a functional unit "dependent upon an instruction type of the instruction." Accordingly, claims 1 and 7 are not made obvious by the combination of Simar et al and Heishi et al.

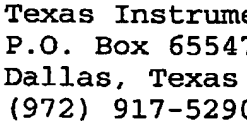
In summary, Heishi et al teaches performing the selection of instructions and dispatch to functional units in a different manner than recited in claims 1 and 7. Heishi et al teaches selectors 224a, 224b, 224c and 224d which determine both the instructions selected and the corresponding functional unit used dependent on their position in the original instruction stream. In contrast, claim 1 recites two structures, multiplexors for instruction selection and dispatch circuitry for functional unit selection, and claim 7 recites two method steps, selecting and dispatching. Accordingly, claims 1 and 7 achieve a similar purpose to the disclosure in Heishi et al employing different structure (claim 1) or different method steps (claim 7).

The Applicants respectfully submit that all the present claims are allowable for the reasons set forth above. Therefore early reconsideration and advance to issue are respectfully requested.

If the Examiner has any questions or other correspondence regarding this application, Applicants request that the Examiner contact Applicants' attorney at the below listed telephone number and address to facilitate prosecution.

Texas Instruments Incorporated
P.O. Box 655474 M/S 3999
Dallas, Texas 75265
(972) 917-5290
Fax: (972) 917-4418

Respectfully submitted,



Robert D. Marshall, Jr.
Reg. No. 28,527